Automated Fluid Feature Extraction from Transient Simulations

Progress Report

Robert Haimes
haimes@orville.mit.edu

January 20, 1998

NASA Ames Research Center

Agreement: NCC2-985

Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139

1 Introduction

In the past, feature extraction and identification were interesting concepts, but not required to understand the underlying physics of a steady flow field. This is because the results of the more traditional tools like iso-surfaces, cuts and streamlines were more interactive and easily abstracted so they could be represented to the investigator. These tools worked and properly conveyed the collected information at the expense of much interaction. For unsteady flow-fields, the investigator does not have the luxury of spending time scanning only one "snap-shot" of the simulation. Automated assistance is required in pointing out areas of potential interest contained within the flow. This must not require a heavy compute burden (the visualization should not significantly slow down the solution procedure for co-processing environments like **pV3**). And methods must be developed to abstract the feature and display it in a manner that physically makes sense.

The following is a list of the important physical phenomena found in transient (and steady-state) fluid flow:

1.1 Shocks

The display of shocks is simple; a shock is a surface in 3-space. As the solution progresses, in an unsteady simulation, the investigator can view the changing shape of the shock surfaces. Some previous work has been done at MIT (as well as other places) on this problem. This early work [Darmofal91a, Darmofal91b] developed the following algorithm:

First determine the normal direction to the shock. Across a shock, the tangential velocity component does not change; thus, the gradient of the speed at a shock is normal to the shock. The exact location of the shock is then determined by calculating the magnitude of the Mach vector, in the direction of the speed gradient, at all points in the domain. The normal Mach number is defined as the Mach vector dotted into the speed gradient. Thus, a positive normal Mach number indicates streamwise compression and a negative normal Mach number indicates expansion. If this value is 1.0 then a shock has been found (or possibly an isentropic recompression through Mach one). This entire iso-surface can be displayed to show the shock, but must be thresholded to remove the surfaces associated with the recompression and some stray portions of the flow field where the normal Mach number happen to be 1.0. The magnitude of the speed gradient was found to be an effective threshold.

1.2 Vortex Cores

Finding these features is important for flow regimes that are vortex dominated (most of which are unsteady) such as flow over delta wings and flow through turbine cascades. Tracking the core can give insight into controlling unsteady lift and fluctuating loadings due to core/surface interactions.

There has been much work done in the location of these features by many investigators. Again, there has been some success [Kenwright97]. This particular algorithm as fully described in [Sujudi95] has been designed so that no serial operations are required, it is parallel, deterministic (with no 'knobs'), and the output is minimal. The method operates on a cell at a time in the domain and disjoint lines are created where the core of swirling flow is found. Only these line segments need to be be displayed, reducing the entire vector field to a tiny amount of data.

This technique, although satisfying, is not without problems. These are:

1. Not producing contiguous lines.
   The method, by its nature, does not produce a contiguous line for the vortex core. This is due to two reasons; (1) for element types that are not tetrahedra the interpolant that describes point location within the cell is not linear. This means that if the core passes through these elements the line can display curvature. By subdividing pyramids, prisms, hexahedra and higher-order elements into tetrahedra for this operation produces a piecewise linear approximation of that curve. And (2) there is no guarantee that the line segments will meet up at shared faces between tetrahedra. This is because the eigenvector associated with the real eigenvalue will not be exactly the same in both neighbors, so when this vector is subtracted form the vector values at the shared nodes each tetrahedra sees a differing velocity field for the face.

2. Locating flow features that are not vortices.
   This method finds patterns of swirling flow (of which a vortex core is the prime example). There are other situations where swirling flow is detected, specifically in the formation of boundary layers. Most implementations of this technique do no process cells that touch solid boundaries to avoid producing line segments in these regions. But this does not always solve the problem. In some cases (where the boundary layer is large in comparison to the mesh spacing) this boundary layer generation is still found.

3. Sensitive to other non-local vector features.

   Critical point theory gives one classification for the flow based on the local flow quantities. 3D points can display a limited number of flow topologies including swirling flow, expansion and compression (with either acceleration or deceleration). The flow outside this local view may be more complex and have aspects of all of these components. The local classification will depend on the strongest type. Also if there are two (strong) axes of swirl, the scheme will indicate a rotation that is a combination of these rotation vectors based on the relative strength of each. This has been reported by [Roth96] where the overall vortex core strength was not much greater that the global curvature of the flow. The result was that the reported core location was displaced from the actual vortex.

1.3 Regions of Recirculation

Recirculation is a difficult feature to locate, but a simple one to visualize. A surface exists that separates the flow (in steady-state) so that no streamlines seeded from one side of this surface penetrate the other side. Some work has been done in locating this feature by computing the stream function. Also it is possible to use vector field topology to find the extent of this region and then draw a series of streamlines connecting the critical points. These lines can be tessellated to create this separation surface.

These methods do not work for transient problems. Like a series of instantaneous streamlines can be misleading in unsteady flow regimes, using techniques based on streamlines will not represent the regions of older fluid. The concept that appears most promising is Residence Time. This is the Eulerian view of unsteady particle tracing (a Lagrangian operation). A simple partial differential equation can be solved on the same mesh along with the flow solver. (NOTE: This is possible when performing co-processing; the CFD solver and Residence Time calculation have the same time limit constraints.) An iso-surface can be generated through the result so that regions of old fluid can be separated from newer fluid elements. Again, some work has been done on this algorithm [Sujudi96].

1.4 Boundary layers

Boundary layers are features that are very important in most complex fluid flow regimes. The size and shape of the boundary layer are used to determine such values as lift and drag in external aerodynamics. For turbomachinery the size of the boundary layers determine the effective solidity. With regions of recirculation, the boundary layers determine the blockage.

In all cases the boundary layer edge can be constructed as a surface (some distance away from solid walls) in 3D flows.

There have been no successes in any known work to robustly determine the surface that represents the extent of the boundary layer from traditional CFD solutions. Fundamentally, this is a very difficult problem. The edge is poorly defined numerically and is more a subtle transition that an abrupt feature.

Accurately knowing the edge of the boundary layer has many numerical benefits for the solver. Turbulence models can be more accurately applied. Grid adaptation can place nodes where they are needed. Split solvers (Euler in core flow, Navier-Stokes in boundary layers) will be more stable and accurate when the position of the edge of the boundary layer is known.

1.5 Wakes

Wakes are usually generated by the merging of boundary layers down stream from a body. Like boundary layers, these features are important for both internal and external flows. Knowing where, and under what circumstances, the wakes impinge on other bodies can have a changing effect on the structural and thermal loads experienced on those surfaces. Again, there has been no real success in finding this feature.

2 Progress This Year

The goal of this work is to develop a comprehensive software feature extraction tool-kit that can be used either directly with CFD-like solvers or with the results of these types of simulations (i.e. data files). The output of the feature "extractors" will be produced in such a manner that it could be rendered within most visualization systems. Much effort will be placed in further quantifying these features so that the results can be applied to grid generation (for refinement based on the features), databases, knowledge based and design systems. This requires two distinct phases; (1) the research into algorithms that will accurately and reliably find these features and (2) the construction of the software tool-kit.

During this first year the following algorithmic work has been accomplished:

2.1 Shocks

The procedure explained above has been re-examined. First, much effort was placed in examining algorithms that find discontinuities in scalar fields. These techniques can be thought of as the 3D analogue to the methods used in image processing. This approach failed in finding shocks for the following reasons:

- Sharpness.
  Most CFD solvers that perform differences to compute derivative and flux quantities do not suppress saw-tooth oscillations in the solution. These can become unstable in even in quiescent flow (for numerical reasons) and will blow-up in the presence of discontinuities. For this reason these CFD solvers "smooth" the flow field. This obviously reduces the ability to find sharp discontinuities since they have been removed. Even for solvers that can handle abrupt changes in the flow field, a shock will probably be smeared across 2 to 3 cells.

- Derivative quantities.
  There tends to be noise generated when derivative quantities are computed from local (cell based) operators. Using operators with larger stencils are possible in structured block meshes but difficult in unstructured grids. This noise problem is amplified when second derivatives are required.

Therefore the shock finder that requires looking for the inflection point – where the laplacian of the laplacian of pressure (the second derivative) is zero is doomed in CFD solutions.

A shock finder has been developed that is a modification of the early work described above. For steady state solutions, the normalized pressure gradient is used instead of the speed gradient – this is less susceptible to other flow features such as boundary layers. It has been found that no thresholding is required. There is also an extension for transient solutions. See the attached document "Shock Detection from CFD Solutions". This working document will be refined and submitted as a paper the IEEE Visualization conference.

Before the end of this contract period, addition work will be done to classify the shocks found as to strength and type (normal, oblique, bow and etc.).

2.2 Vortex Cores

The current algorithm produces a series of disjoint line segments. When displayed, the eye puts together (or closes) a single line, for a single core, (when the strength of the core is large). This is not acceptable for off-line uses (the first problem listed above) in that it is not possible to trace the full extent of the core. Work is underway that resolves this issue. Enforcing the cell piercing to match at cell faces insures that the line segments generated will produce a contiguous core. This can be done via the following modification to the algoithm:

1. Compute the Velocity Gradient tensor at each node.
   This requires much more storage – 9 words are needed for each node in the flow field. This has the advantage that the stencil used for the operation is larger than the cell and therefore the result will be generally smoother.

2. Average the node tensors (on the face) to produce a face-based Velocity Gradient tensor.
   This insures that the same tensor is produced for the two cells touching the face.

3. Perform the eigen-mode analysis on the face tensor.
   If the system signifies swirling flow, determine if the swirling axis cuts through the face by the scheme used in the current method. If, so mark the location on the face.

This scheme will work at the expense of memory and a much higher CPU load. Four eigen-mode calculations are required for each tetrahedron instead of just one. In general, this can be reduced to two per tetrahedron, by the additional storage of face results (about 3 words per face). Note: there are about 2 times the number of faces as cells in a tetrahedral mesh.

This is not a good result, in particular for structured blocks, where each individual hexahedron is broken up into 6 tetrahedra (5, the minimum does not promote face matching). This means that for each element in the mesh a minimum of 12 eigen-mode analyses are required.

These performance problems suggest another, related, technique:

1. Compute the Velocity Gradient tensor at each node.

2. Perform the eigen-mode analysis on the node tensor.
   The tensor can be overwritten with the critical point classification and the swirl axis vector for rotating flow.

3. Average the swirl axis vectors for the nodes that support the tetrahedral face.
   This should only be done if all nodes on the face indicate swirling flow. Some care needs to be taken to insure that the sense of the vectors are the same. Determine if the swirling axis cuts through the face, and if so, mark the location on the face.

This shows great potential. For tetrahedral meshes, the reduction of compute load is by a factor of 5 to 6 over the original method (there are roughly 5.5 tetrahedra per node in 'good' unstructured grids). For structured blocks, where the number of nodes is about equal to the number of hexahedra, the number of eigen-mode analyses required is on the order of one per cell.

Before the end of this contract period, this new scheme will be implemented and tested.

2.3 Boundary layers and Wakes

Some progress has been made in this difficult arena. An algorithm is being constructed that will allow the use of iso-surfacing to separate the boundary layers and wakes from core flow. The method stems from the fact that these features display both rotating flow and fluid under shear stress. This is why, sometimes the vortex core technique gives false-positives for locations in boundary layers. Therefore, with a boundary layer finder we should be able to mask out these finds in the boundary layer and only display those lines that trace back from the outer flow.

To numerically define these quantities we again start with the Velocity Gradient tensor at each node:

- Rate of Rotation.

  This quantity is related to vorticity. A skew-symmetric tensor is produced by subtracting the transpose of the Velocity Gradient tensor from the Velocity Gradient tensor. The result has zero on all of the diagonal terms and the off-diagonal terms are symmetric but have opposite signs across the diagonal. These values are coordinate system invariant. For this application, the norm of the upper (or lower) terms is used for the rotation scalar. This is a measure of the rate of solid-body rotation.

- Rate of Shear Stress.

  A symmetric tensor can be produced from the Velocity Gradient tensor by adding it to its transpose. This defines the Rate of Deformation tensor. The matrix represents both the bulk and shear stresses and is dependent on the coordinate system. To extract a single scalar that is coordinate system invariant and has the bulk terms removed it is necessary to diagonalize this tensor. The result produces a vector which signifies the 'principle axis of deformation'. By employing techniques from Solid Mechanics, the norm of the second principal invariant of the 'stress deviator' can be used as a measure of the shear and employed as the scalar.

Currently, (and for the lack of anything better), a node based scalar field is produced that is the product of the shear scalar and the rotation scalar. This has the proper characteristics that both shear and rotation are required to mark the node as being in the boundary layer/wake region. In fact, the square root of this quantity is actually used in order to preserve the units of inverse time.

Figures 1 and 2 show an iso-surface of this quantity in 3D flow fields.

3 Presentations and Publications

Through the paper and video presented at the 1997 IEEE Visualization conference [Kenwright91] was not funded from this contract, it is germane to the work. The paper discussed a number of applications of the original vortex core technique. This was awarded 'Best Case-Study'.

In conjunction with the conference SuperComputing '97 on the 19 of November 1997, a Birds-Of-A-Feather on automated feature extraction was held at NASA Ames Research Center. The purpose of this discussion was to get the industry, NASA and Army personnel interested in this topic together at one location. An overall presentation was given (by Robert Haimes) on the direction of this work. Dave Kenwright (NASA Ames) talked about the vortex work – the paper given at the Visualization conference and some 2D extensions for finding separation lines on body surfaces. Dave Lovely (an MIT Student) talked about the beginnings of the algorithm work on the shock locator. Finally, the discussion was open to the attendees. The topics were; (1) are these the correct features, (2) the kind of output desired for each feature and (3) what is important (the priority).

4 Next Years Effort

4.1 Regions of Recirculation

The recirculation algorithm described above needs to be closely integrated with the flow solver in some way. The choice is either that solver writer completely incorporates this by adding one more equation to the state-vector or some co-processing system (like the visualization suite **pV3**) is used. Obviously, the best place for this PDE is within the solver, in particular when there has been partitioning. This is because of the time-step constraints of the Residence Time equation (the same as the solver) in conjunction with the method used for integration in time, and the updating of information with the partition (and other) boundaries. For the second choice, an API for solving this PDE will be developed so that there is access to all of the required data. A Lax-Wendroff scheme will be used for the time integration, therefore if some implicit or high-order explicit time integration scheme is used for the solver care must be taken in selecting the time-step so that the solving of the Residence Time equation is stable.

4.2 Boundary layers and Wakes

The current scheme shows promise but it has the following problems:

- The function of shear and rotation is currently *ad hoc*.
- The value is not non-dimensional, but has units of inverse time.
  This means that the iso-surface value used to define the edge of the layer changes from case to case. This scalar needs to be multiplied by some characteristic time associated with the problem.
- The value used for the iso-surface is not specified via theory.

The work next year will focus on resolving these problems. This will include a rigorous approach with either theory or computations of isolated flat-plates (where the theory exists) or both.

4.3 Vortex Cores

The new scheme outlined here will be enhanced so that coherent lines (instead of line segments) are produced. This will be done by collecting the line segments and tracing them through the flow. The boundary layer work will be used to suppress the identification of cores within this region unless swirling flow persists outside.

A measure of core strength will be found that can be mapped onto the core line or integrated to get a single measure.

5 References

Darmofal91a David Darmofal, "Hierarchal Visualization of Three-Dimensional Vortical Flow Calculations". MIT Thesis and CFDL-TR-91-2, March 1991.

Darmofal91b David Darmofal and Robert Haimes, "Visual Feature Identification for 3-D Data Sets". AIAA Paper 91-1583, Honolulu HI, June 1991.

Kenwright97 David Kenwright and Robert Haimes, "Vortex Indetification – Applications in Aerodynamics". IEEE Computer Society, Visualization '97, Oct. 1997. Awarded 'Best Case-Study'.

Sujudi95 David Sujudi and Robert Haimes, "Identification of Swirling Flow in 3-D Vector Fields". AIAA Paper 95-1715, San Diego CA, June 1995.

Rort96 M. Roth and R. Peikert, "Flow Visualization for Turbomachinery Design". IEEE Computer Society, Visualization '96, Oct. 1996.

Sujudi96 David Sujudi, "Distributed Visualization and Feature Identification for 3D Steady and Transient Flow Fields". MIT Thesis, February 1996.

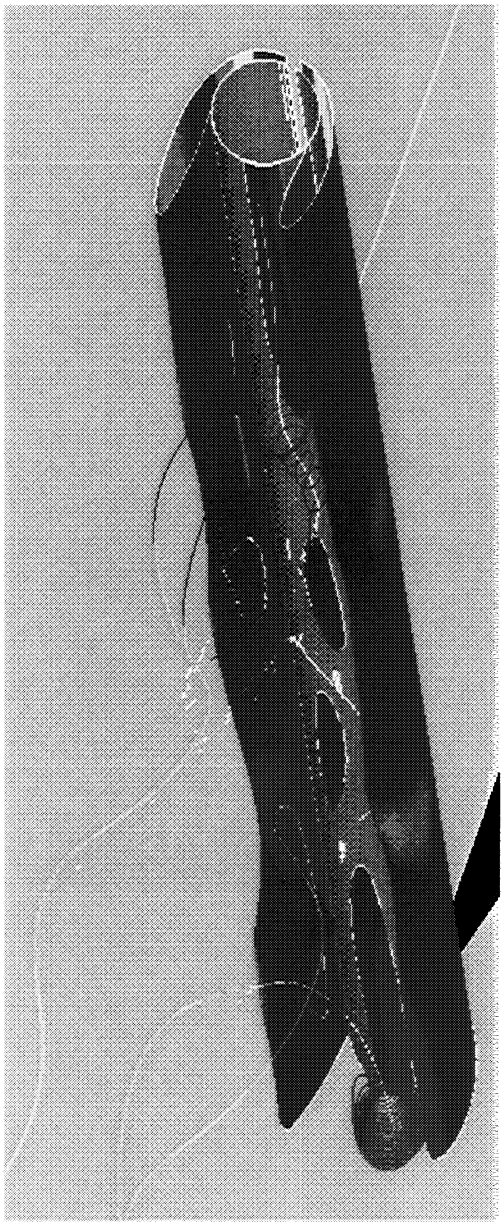


Figure 1. Flow behind a tapered cylinder.

This image depicts the boundary layer function (the colored iso-surface) as seen behind the flow about a tapered cylinder. The white disjoint line segemnts represent the results from the vortex core finder. The colored lines are seeded streamlines. The color represents density where the minumum (blue) is 0.9567 and the maximum (red) is 1.0192.

This data is from:
Dennis Jespersen and Creon Levit, "Numerical Simulation of Flow Past A Tapered Cylinder". AIAA paper 91-0751, January, 1991.

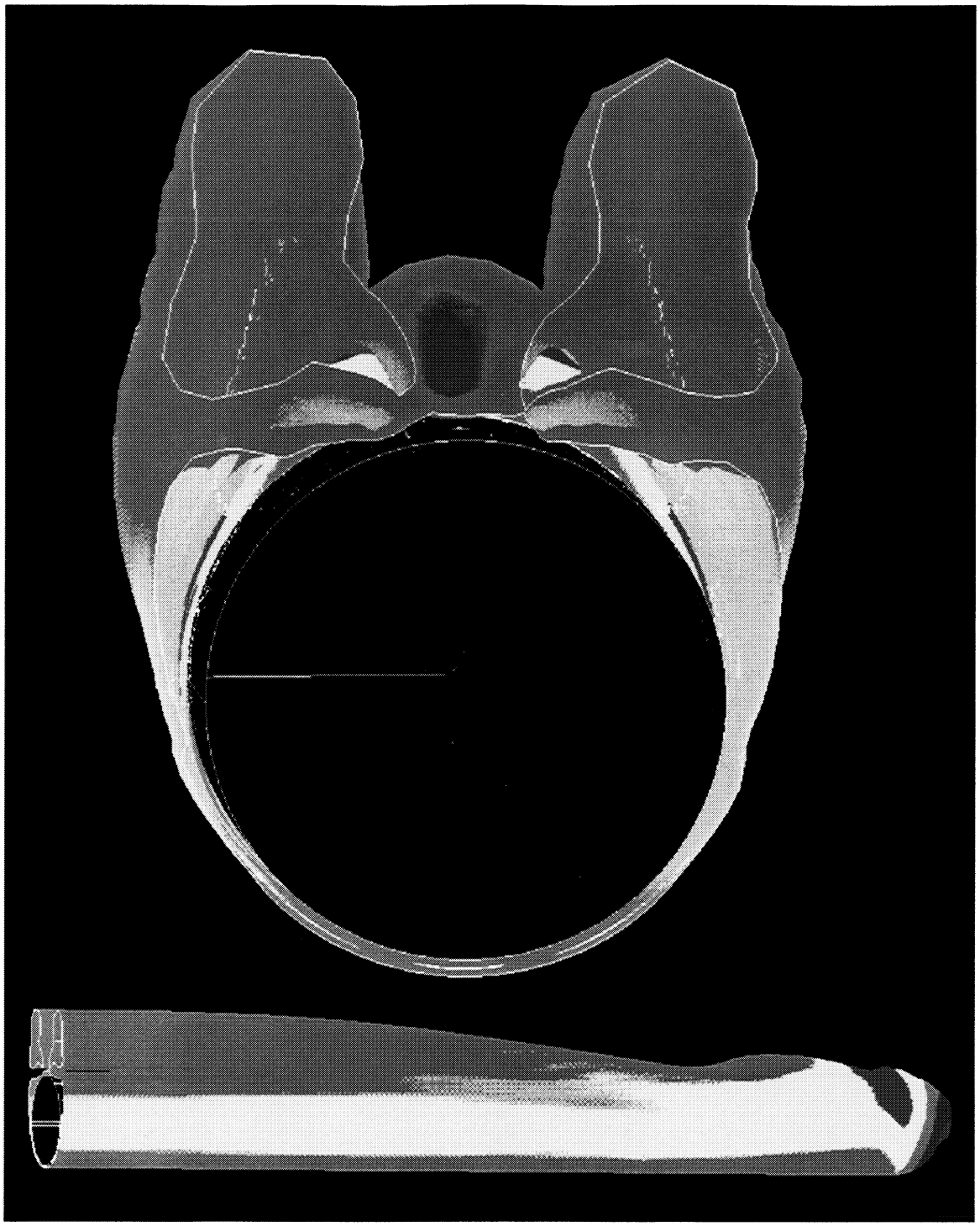


Figure 2. Flow about a hemisphere cylinder.

This picture shows two views of the boundary layer function (the colored iso-surface) computed from the solution about a hemisphere cylinder. The flow is incoming at 5 degrees off the cylinder's axis and the Reynolds number is 14,000. The white disjoint line segemnts (seen in the top image) represent the results from the vortex core finder. The color represents Mach number where the minumum (blue) is 0.0 and the maximum (red) is 2.0774.
Note: in this case the extent of the vortex core is caught by this iso-surface.

This data is from:
Thierry Delmarcelle and Lambertus Hesselink, "Visualization of Second Order Tensor Fields and Matrix Data". IEEE Visualization '92, September 1992.

Shock detection from CFD solutions

MIT Aero Astro
January 20, 1998

David Lovely

Summary

An algorithm is introduced to locate and display areas of shock in CFD solution fields. The algorithm is tested for a variety of shock types and different solution fineness. The effectiveness of the algorithm is tested on CFD solutions to the classical problem of a sharp wedge in supersonic flow. Three separate grids were used to test the sensitivity of the algorithm to solution fineness. The results indicated that the algorithm was able to locate the oblique shock, and displayed it as a region, with a much larger thickness than a physical shock. This was found to be a result of numerical model used to approximate the discontinuity, and was effected by the element size. The smaller the elements, the higher the gradients, and the smaller the shock region.

A couple of correction terms were added to locate transient shocks. Then the ability of the algorithm to find moving shocks was tested on another classical problem, a one dimensional shock traveling in a steady flow. The algorithm proved to be successful in finding these shocks. but also displayed some false shocks caused by the CFD solver. These false shock indications could be removed by only displaying a shock when the magnitude of the pressure gradient reached a certain threshold. A heuristic algorithm was introduced to find a pressure gradient magnitude threshold to filter out all the false shocks.

Related Work

Shock waves are discontinuities in the flow fields that may occur when the velocity of the fluid exceeds the speed of sound. The state of the fluid as described by the pressure, velocity and other primitive variables can change radically across a shock boundary of only a few molecular paths wide. However, when the flow is numerically modeled, the shock feature is often smoothed out over a greater distance, and the discontinuity is not as pronounced. Further, it becomes difficult to recognize where the shock occurs by only looking at the primary variables that are output from the numerical model. For example, the pressure may change dramatically across a shock, but the reverse is not often true. So test quantities are calculated from these primitive results, and when the value of this new variable exceeds some threshold, a shock is indicated. The difficulty is to produce a shock variable that is accurate enough to capture all the regions where a shock may be occurring and exclude areas where related flow phenomenon may be occurring, like expansion waves.

In the past, shock waves have been extracted from data sets with a couple of techniques. The first is to look for inflexion points in the pressure or density fields and threshold out those areas with small gradients. At a shock, the pressure gradients go through a change of 180 degrees in direction and a magnitude change as well. At a shock the second derivative of the pressure goes to zero, as shown in the following figure. However, this also occurs in quiescent flow, so it is necessary to filter out those areas with small pressure gradients.

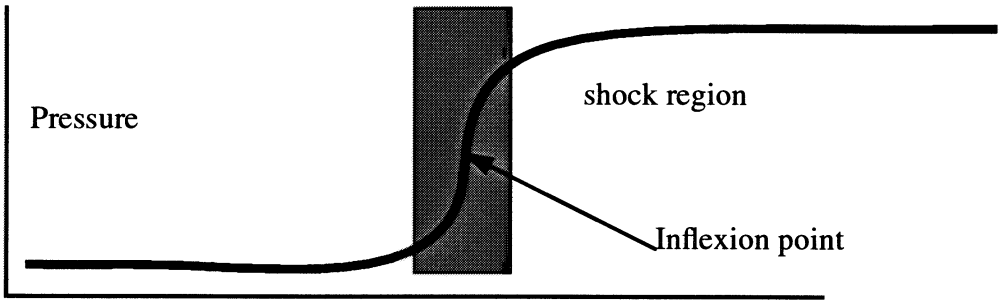


figure 1: Shock inflexion points

A second method is the one used in this paper, which is to use the pressure or density gradients to find the value of the mach number normal to a shock. A shock is then located where this normal mach number exceeds one. This algorithm is described in more detail in the following section.

Stationary Shocks

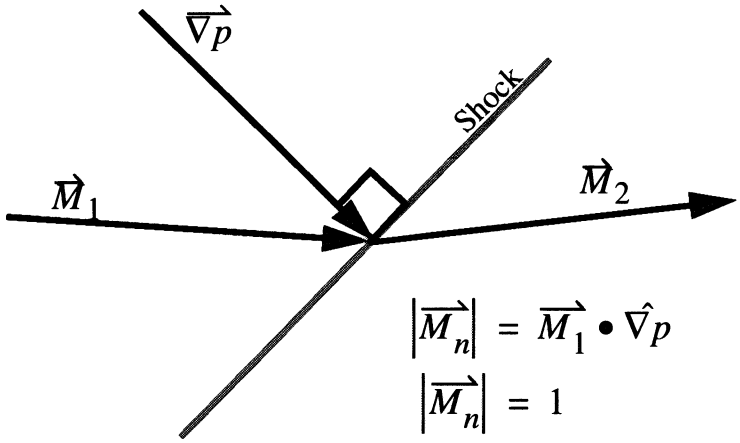
There are a couple of questions that need to be answered for any feature detection algorithm. Is it accurate, i.e. does it detect features that are actually present in the data. And secondly, does it exclude all other features. To answer these questions, a couple of case studies were done, comparing solutions over different grids to see if the features that the algorithm found match the theoretical or experimental location of the features.

The stationary shock algorithm was developed by knowing something about the shock geometry, which is shown in the following figure. For any shock, the mach number normal to the shock has a value of one just before the shock. This normal mach number can be computed on each node and used as a test value for determining the shock location. To compute the normal mach number, to see if a shock occurred, it is necessary to find what the shock orientation would be. The pressure gradient can be used to find the shock orientation because it is always normal to the shock. So, the pressure gradient was approximated for each node, and then used along with the mach vector to calculate a shock test value at each node. Where the test value equals one forms a boundary surrounding the shock location.

figure 2: Shock detection test quantity

When applied to three dimensional models, an isosurface was created where the normal mach number equaled one. The shock feature is surrounded by the M normal = 1 isosurface, and has a thickness associated with it. In the two dimensional case, contours of the normal mach number were created, and the Mn = 1 curve forms a boundary for a shock region in the two dimensional model.

Numerically, there is a difficulty with the normalized pressure gradient in some cases. When the magnitude of the pressure gradient is zero, the calculation for the normalized pressure gradient has a division by zero error. To get around this, a small number can be added to the magnitude of the pressure gradient. Then when the components of the pressure gradient are multiplied by one over the magnitude of the pressure gradient, the numbers will not be unreasonably large. An alternative method, which was used in this experiment, is to set the normal mach number to



zero when the magnitude of the pressure gradient got below a certain threshold.

Shock strength

Beside the location of a shock, it is also of interest to find the relative strength of the shock. Ideally, this could be calculated by the ratio of some invariant flow variable across the shock. For example, P2/P1 would be a good measure of the shock strength. It has to be an invariant variable, because plotting velocity or mach ratios will give incorrect results when the shock is translating. Calculating P2/P1 is a problem for two reasons. The first is that it is necessary to find the exact extent of the shock region with a contouring algorithm in the two dimensional case, or an isosurfacing algorithm in the three dimensional case. It's not a quantity that can be calculated on a node by node basis, but only with knowledge of the interpolants used in the elements. This is just computationally a bit more difficult, but the second problem is more fundamental. The shock finding algorithm relies on high gradients to mark the shock region. However, near the boundaries of the shock, the pressure and density gradients get smoothed due to viscosity in the real shocks and dissipation in numerical shocks. The result is that the detected shock region will not encompass then entire shock. This means that the pressure ratio across the boundary of the detected shock region will be smaller than the actual pressure ratio across the shock. Later in this paper, tests were done to see if the pressure ratio across the detected region could be used to calculate the actual pressure ratio with an empirical relation.

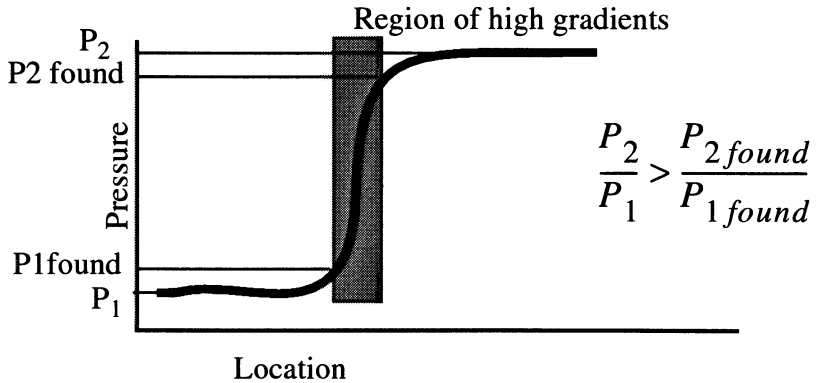


figure 3: Computed vs. actual pressure ratio

Shock Classification

Results

Supersonic ramp

The supersonic ramp test case had the geometry detailed in the following figure. The goal was to see if the shock detector would show clear oblique shocks and reject areas of expansion. A test case was also run with a mach number that would not support an oblique shock to see if the detector would capture these detached shocks.

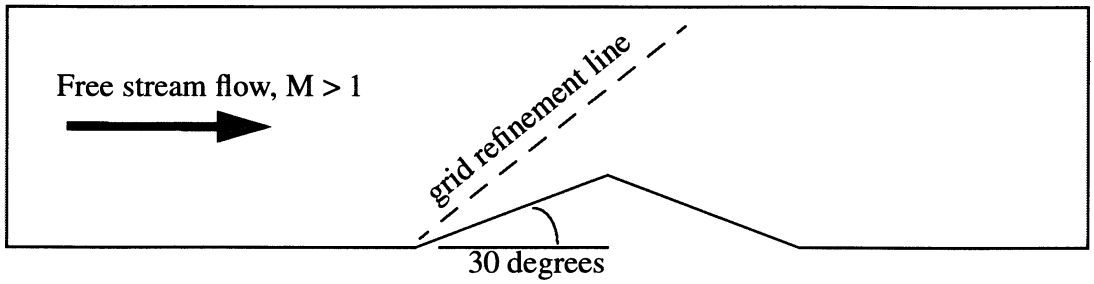


figure 4: Ramp model

Oblique shock

The first case was run at mach three on a relatively fine grid, which theoretically should produce an oblique shock of 45 degrees. The following figure shows the grid that was used.

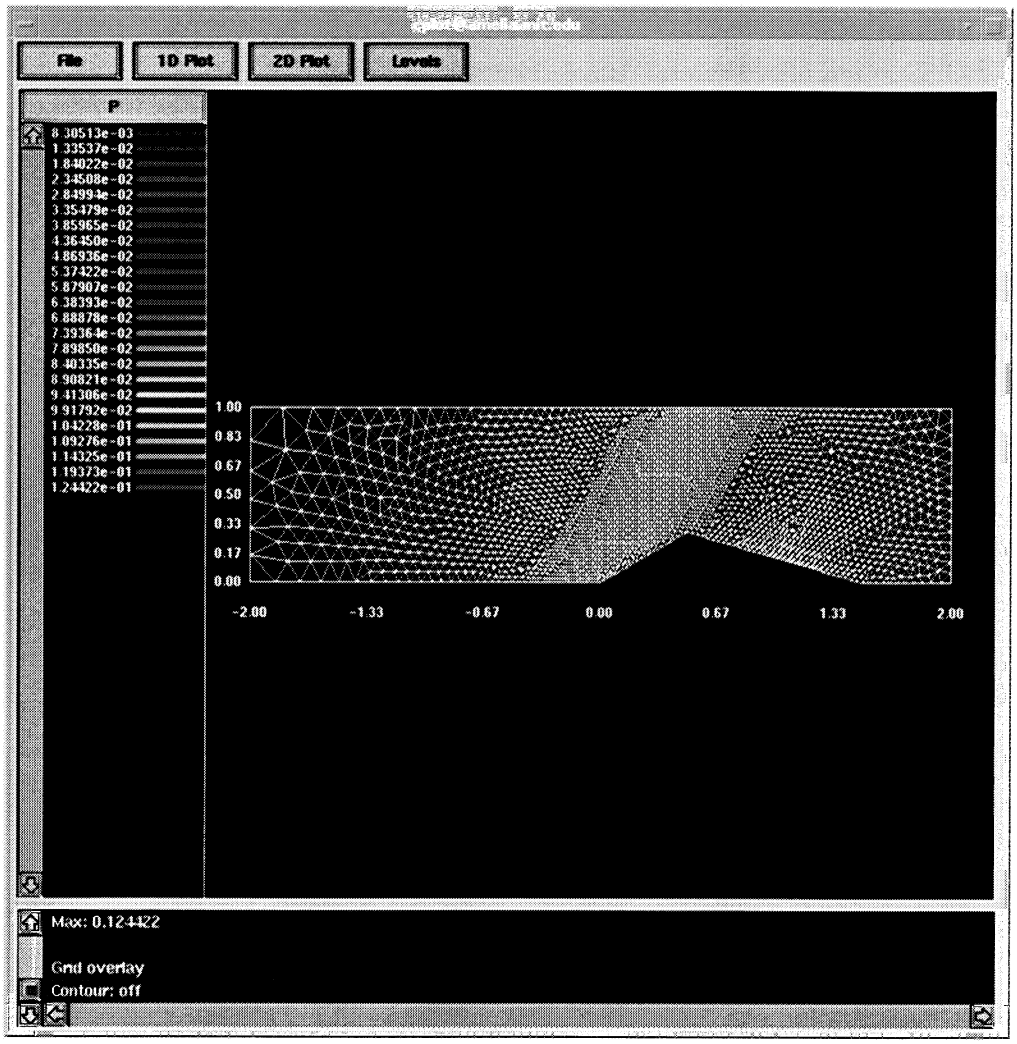


figure 5: Fine wedge grid

In this case, the shock location can be effectively inferred from the pressure contours shown in the following figure. The location is determined by looking for areas of close pressure contours. This region in the front of the wedge corresponds nicely to an oblique shock of 45 degrees, as predicted in invicid compressible flow theory. An even better estimate can be made by looking at the pressure gradient contours shown in figure 6. Finally, figure 7 shows the shock value contours. Note that this is very similar to the pressure gradient contour plot, except that the areas of flow expansion and small pressure gradients have been effectively filtered out.

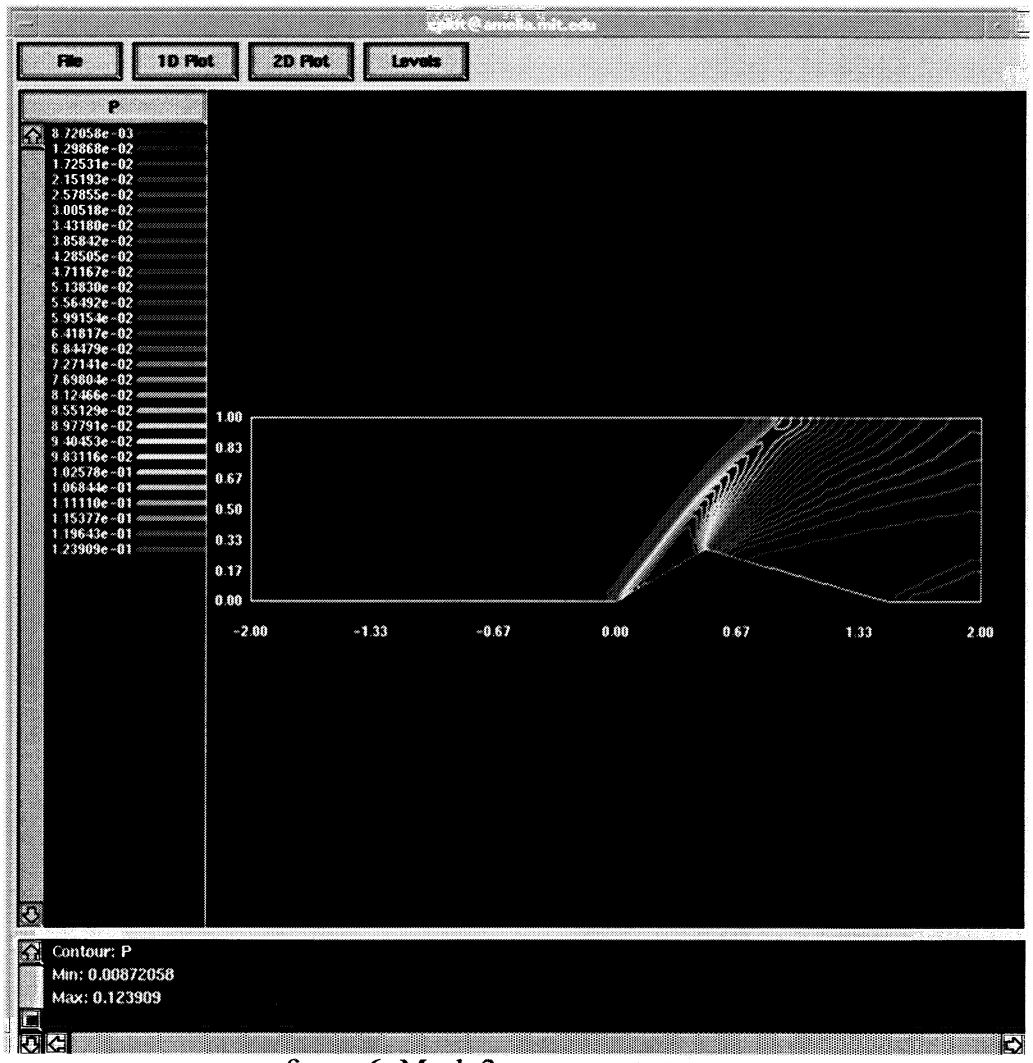


figure 6: Mach 3 pressure contours

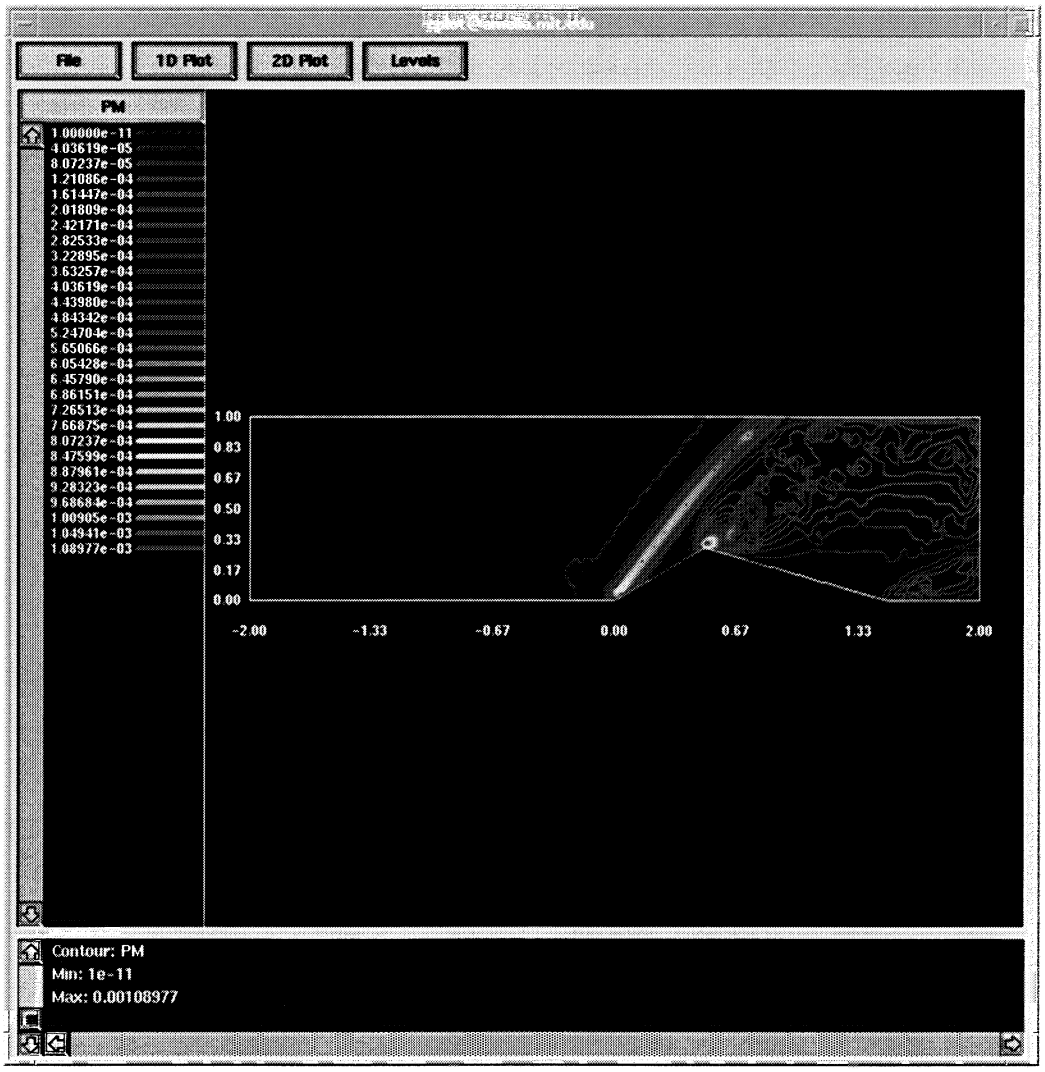


figure 7: Contour plot of the pressure gradient magnitude

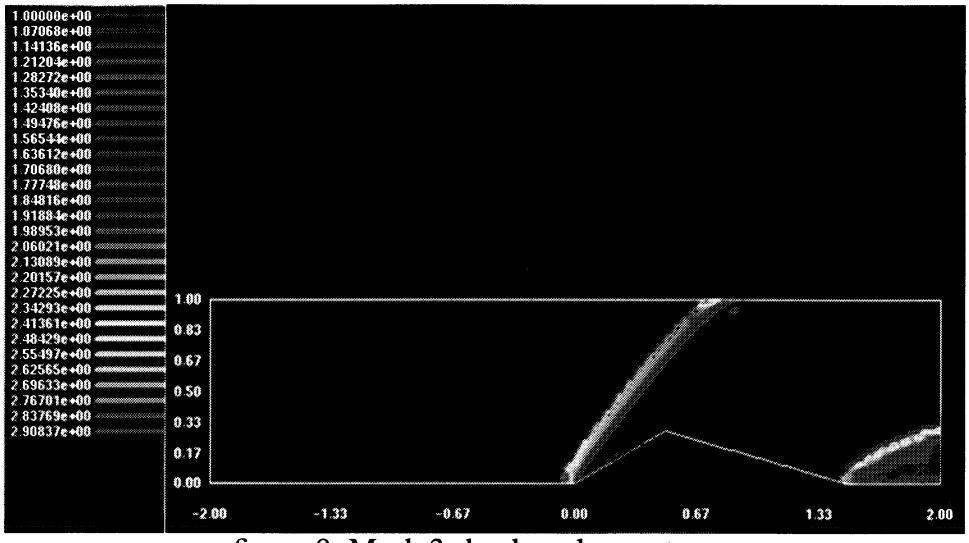


figure 8: Mach 3 shock scalar contours

Because the shock scalar quantity is calculated with a derivative, it is more sensitive to a poor quality solution than is a primary quantity like pressure. So, the shock contours will not only show the location of the shock, but also the quality of the solution better than contours of the primary variables.

Grid study

The same model was run with a couple of other grids. The first was relatively coarse, and the second was very refined. The following figure shows the grid that was used in this experiment. Note that some refinement was still used in the area of the shock

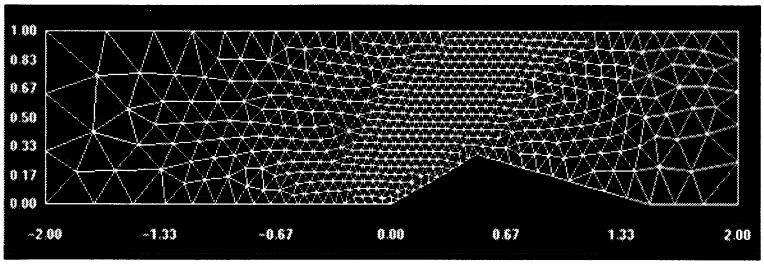


figure 9: wedge coarse grid

Figures 9 and 10 show the results that were obtained on this coarse grid. The were similar results, except that the pressure contours were spaced further apart, indicating that the pressure gradients were smaller for the coarse grid. This had the effect of making the detected shock region larger than on the coarse grid. The thickness of the indicated shock is dependent on the fineness of the grid.

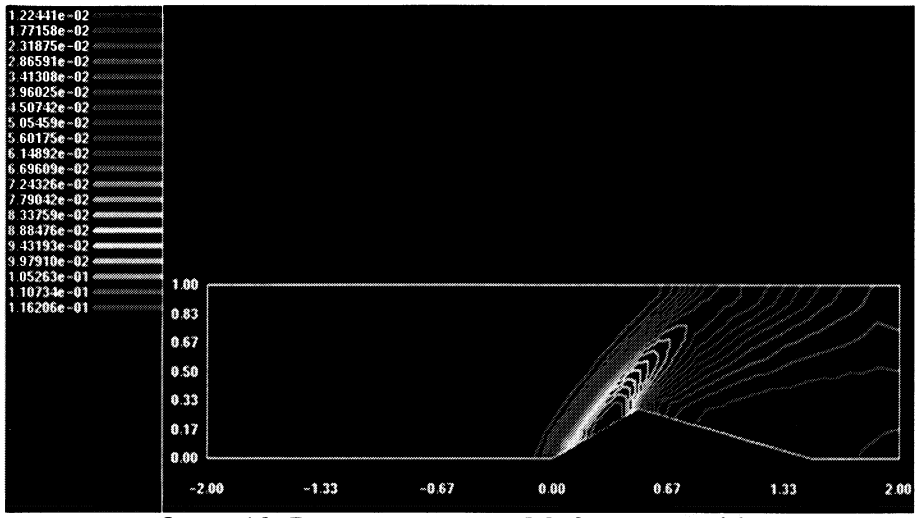


figure 10: Pressure contours M=3, coarse grid

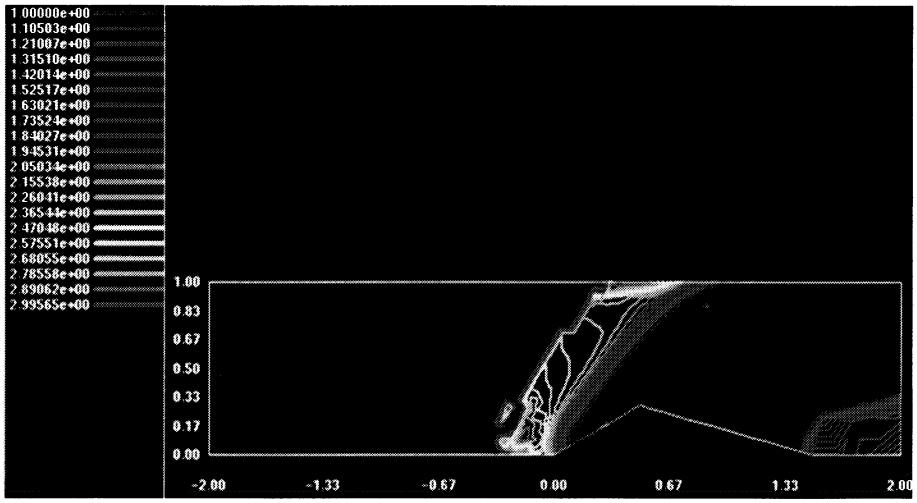


figure 11: Shock value contours M=3, coarse grid

On the very fine grid, an interesting phenomenon occurred with the pressure contours. that has an impact on the shock finding algorithm. The following figure is a graph of the pressure along a line normal to the shock. The main pressure discontinuity is clearly shown by the figure,

but so is a high frequency pressure oscillation after the shock.

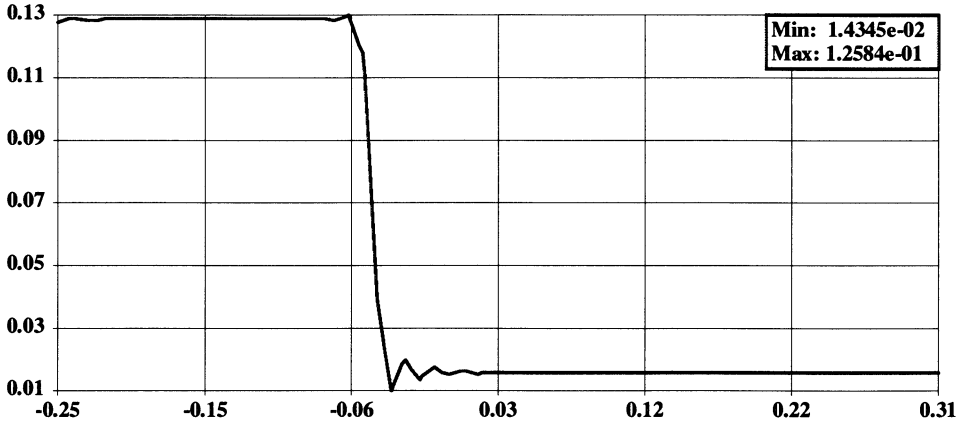


figure 12: Pressure vs. location across the shock

The high frequency pressure oscillation effects the shock finding algorithm because these oscillations produce relatively high pressure gradients, which get picked up by the algorithm and displayed as a shock. The following figure shows the result, the shock contours have isolated islands away from the actual shock. This problem can be resolved by thresholding out all areas with a small pressure gradient magnitude. Similar problems show up in the transient shocks, and a method for dealing with them is presented in this section.

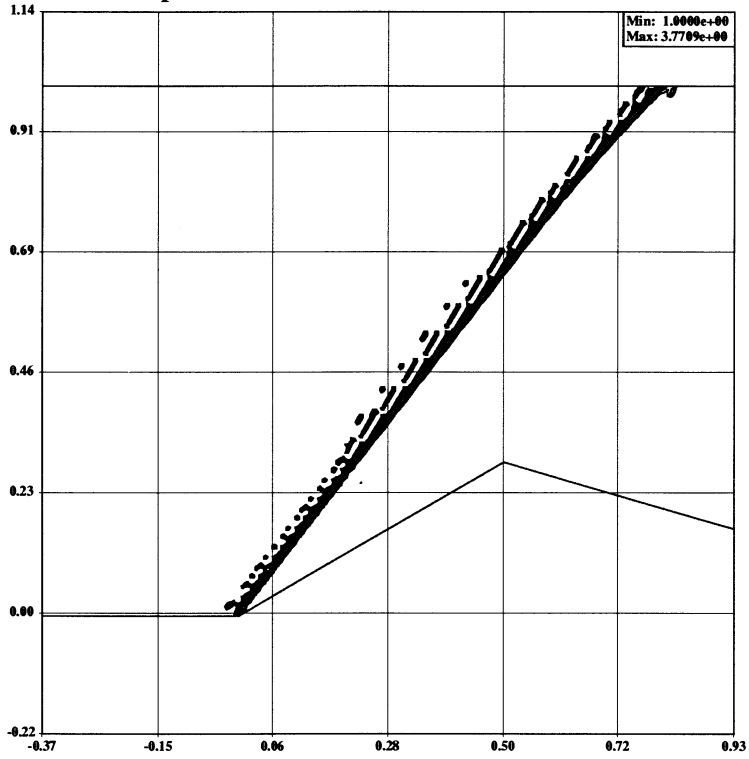


figure 13: Shock contours on very fine mesh.

The results that are shown in the following table show that the shock algorithm finds more defined shocks with increasing numbers of elements. So, the shape of the shock region can not

only locate the shock, but point to a lack of mesh refinement in the area.

Table 1: Grid study results

Number of elements	Shock thickness
1129	NA
8014	0.12
28046	0.05

The grid was also modified to see if alignment of the elements would affect the shock detection. The elements were aligned so that one edge followed the shock by placing the line-source directly on the shock. The results shown in the following figure is that alignment o f the elements does effect the edges of the shock boundary. Regular, aligned elements seem to produce jagged shock contours.

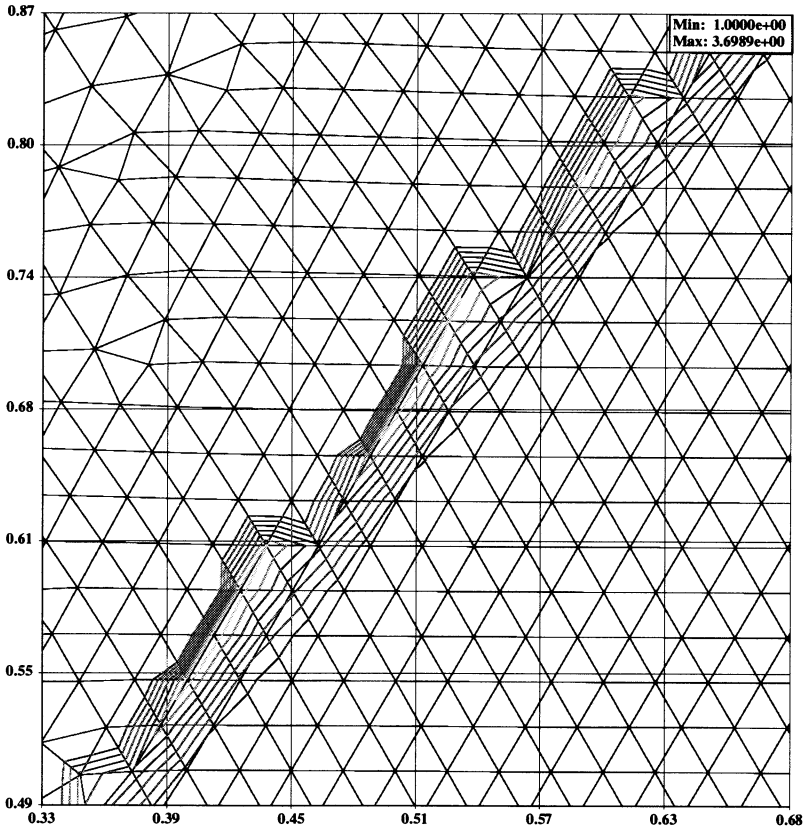


figure 14: Shock contours and on a regular grid

Detached shock

Another run of the same model was made at a lower mach number to see if the algorithm could capture detached shocks. If the pressure contours shown in the following figure were used to determine the shock location, the wrong conclusions would be reached. From the figure, it

looks like there is a sharp oblique shock that is attached to the ramp. However, when the shock finder is applied (figure 3), it is clear that a shock does not occur in this area, which is consistent with the theory.

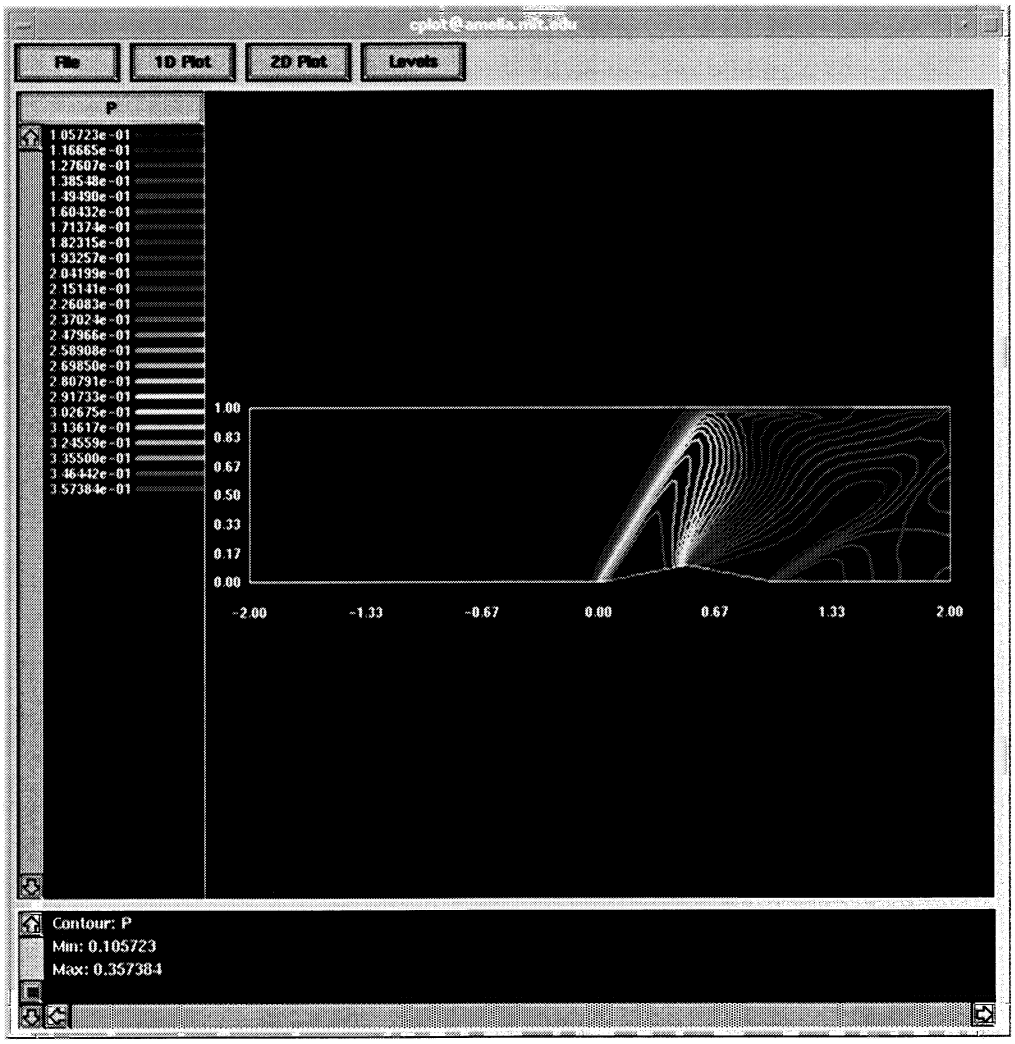


figure 15: Pressure distribution of wedge in M=2 flow

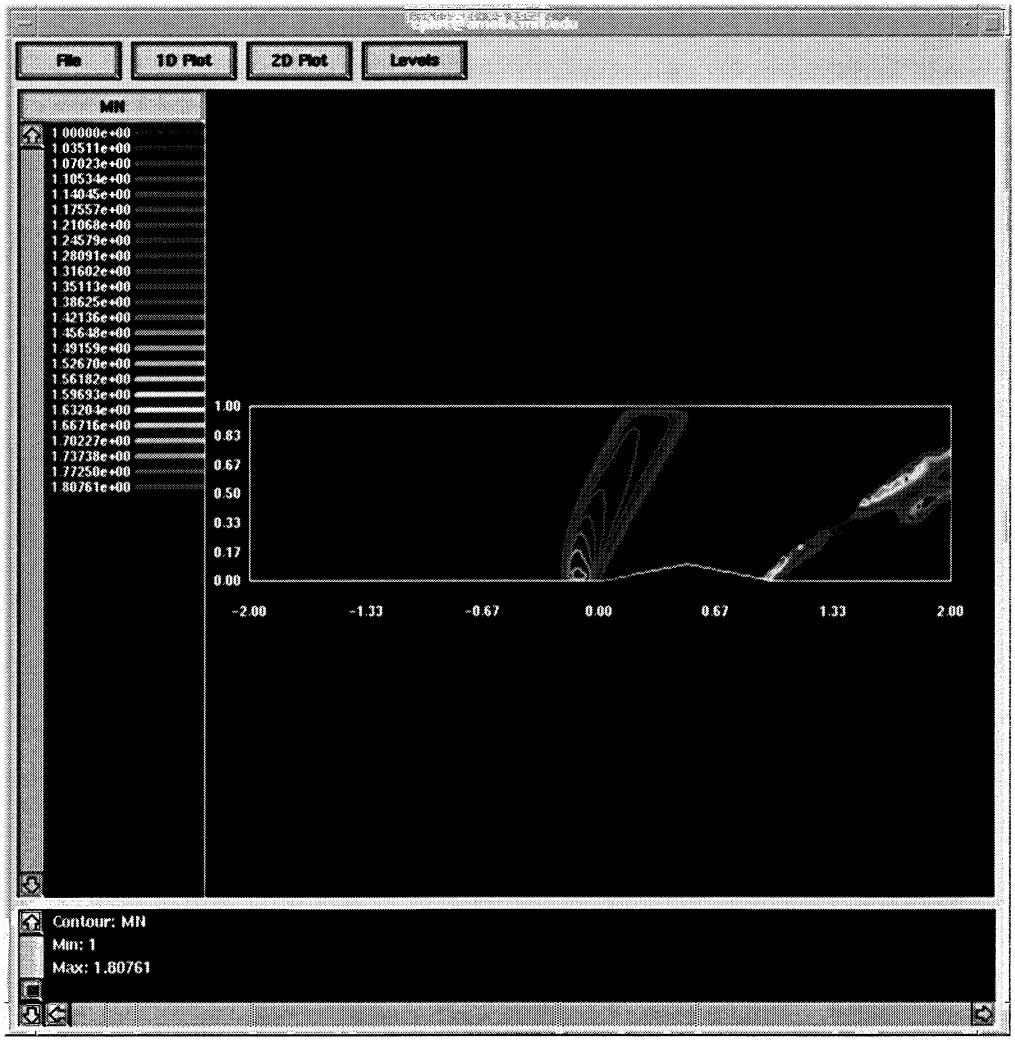


figure 16: Shock scalar distribution of wedge in M=2 flow

So, the shock finding algorithm can give information about the nature of the flow, without showing clear, well defined shock surfaces.

Discussion

In the two dimensional case, the shock is depicted as a region of high gradients, not as the thin discontinuity that it really is in nature. In three dimensions, when an isosurface is located where the shock scalar equals 1, a similar three dimensional region is enclosed, and it often looks as if the shock has two surfaces. In the past, researchers have attempted to filter out one of these surfaces by weighting the nodes by pressure or density gradient. Then only the surface with the largest gradients will show up in the final display. However, the fact that there is a shock region is a useful thing. The size of the region points to problems of grid resolution, where there is a coarse grid, the gradients tend to be smeared out, causing the shock feature to be less well resolved.

Transient Corrections

The assumptions made for finding the location of the shock no longer apply when the shock is moving. The problem is that the normal mach number across the shock will be different from one if the shock is moving. There has to be a correction applied to account with the moving frame of reference. The following equation shows what this term must be, basically a time derivative of the pressure.

$$\frac{1}{|\nabla p|}\frac{1}{a}\frac{Dp}{Dt} = \frac{1}{|\nabla p|}\frac{1}{a}\frac{\partial p}{\partial t} + \vec{M}\cdot\hat{\nabla} p$$

It is more computationally expensive to approximate time derivatives directly, since that would require the storage of multiple time steps. So, the time derivative of pressure was calculated based on relations that equate it to a spacial variation of the state variables. The first equation applies to isentropic flows.

$$\partial p = a^2 \partial \rho$$

This equation is then used along with the conservation of mass equation to produce an equation for an invariant test quantity that can be used to locate moving shocks.

$$\frac{\partial \rho}{\partial t} = -\nabla\bullet(\rho\vec{q})$$

$$\frac{1}{|\nabla p|}\frac{1}{a}\frac{Dp}{Dt} = -a\frac{1}{|\nabla p|}\nabla\bullet(\rho\vec{q}) + \vec{M}\cdot\hat{\nabla} p$$

A shock is then located when this quantity equals 1.

In the general case, pressure can be related to the internal energy and velocity of the flow.

$$p = (\gamma - 1)\left[\rho E - \frac{1}{2\rho}(\rho\vec{q})\bullet(\rho\vec{q})\right]$$

$$\frac{\partial p}{\partial t} = (\gamma - 1)\left[\frac{d}{dt}(\rho E) - \vec{q} \bullet (\rho \vec{q}) + \frac{1}{2} q^2 \frac{\partial \rho}{\partial t}\right]$$

$$\frac{\partial p}{\partial t} = (\gamma - 1)\left[-\nabla\bullet(\rho \vec{q} H) + \vec{q} \bullet (\nabla p + \nabla\bullet\rho \vec{q}\vec{q}) - \frac{1}{2} q^2 \nabla\bullet(\rho \vec{q})\right]$$

$$\frac{1}{p}\frac{1}{a}\frac{Dp}{Dt} = \frac{1}{|\nabla p|}\frac{1}{a}(\gamma - 1)\left[-\nabla\bullet(\rho \vec{q} H) + \vec{q} \bullet (\nabla p + \nabla\bullet\rho \vec{q}\vec{q}) - \frac{1}{2} q^2 \nabla\bullet(\rho \vec{q})\right] + \vec{M} \cdot \nabla$$

Shock Speed

The speed of the moving shock wave can be approximated with by looking at the magnitude of the correction term.

Test Case

Translating normal shock in a tube

A model of a moving normal shock in a channel was created to investigate the behavior of the shock finding algorithm, and the effect of the transient modification. Two separate runs were done, the first had the following initial conditions shown in the following figure.

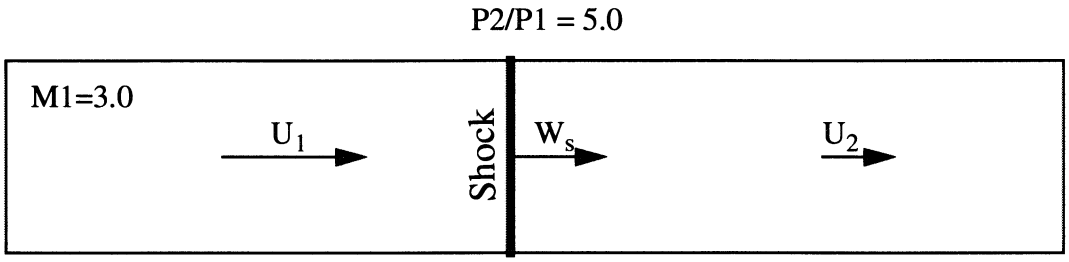


figure 17: Transient shock model

The formulas for moving normal shock waves with constant Cp and Cv are applicable in this case. However, these formula assume that the upstream velocity, U_1 is 0, so a correction had

to be made to produce the correct initial conditions of the flow. For the first run, the pressure ratio was chosen to be less than the pressure ratio for a standing shock in M=3 flow (10.33). This required that the shock move toward the right. The speed of the shock traveling into stationary flow was calculated with the following formula.

$$W = a_1 \sqrt{\frac{\gamma+1}{2\gamma}\left(\frac{P_2}{P_1}-1\right)+1}$$

Then the speed of the flow behind the a shock traveling into stationary flow was calculated with the following formula, which required the density ratio across the shock.

$$U_p = W\left(1-\frac{\rho_1}{\rho_2}\right)$$

This ratio was calculated with another shock relation, and is only dependent on gamma and the pressure ratio.

$$\frac{\rho_2}{\rho_1} = \frac{1+\frac{\gamma+1}{\gamma-1}\left(\frac{P_2}{P_1}\right)}{\frac{\gamma+1}{\gamma-1}+\frac{P_2}{P_1}}$$

Since the upstream velocity was not zero, the actual speed of the shock had to be corrected with the following formula.

$$W_s = U_1 - W$$

Similarly, the downstream velocity was determined by subtracting the change in velocity due to the shock, Up from the upstream velocity U1.

$$U_2 = U_1 - U_p$$

The results are shown in the following figures. The pressure distribution across the shock has some interesting features that are a problem for the shock finding algorithm. The shock started at X=0, and is moving to the right. As it moves, the shorter wavelengths that makeup the initial discontinuity move at a slower speed than the longer wavelengths. This difference in speed is a numerical artifact of the time stepping method used in the CFD solver. So, high frequency pres-

sure oscillations show up behind a moving shock, as shown in the following figure.

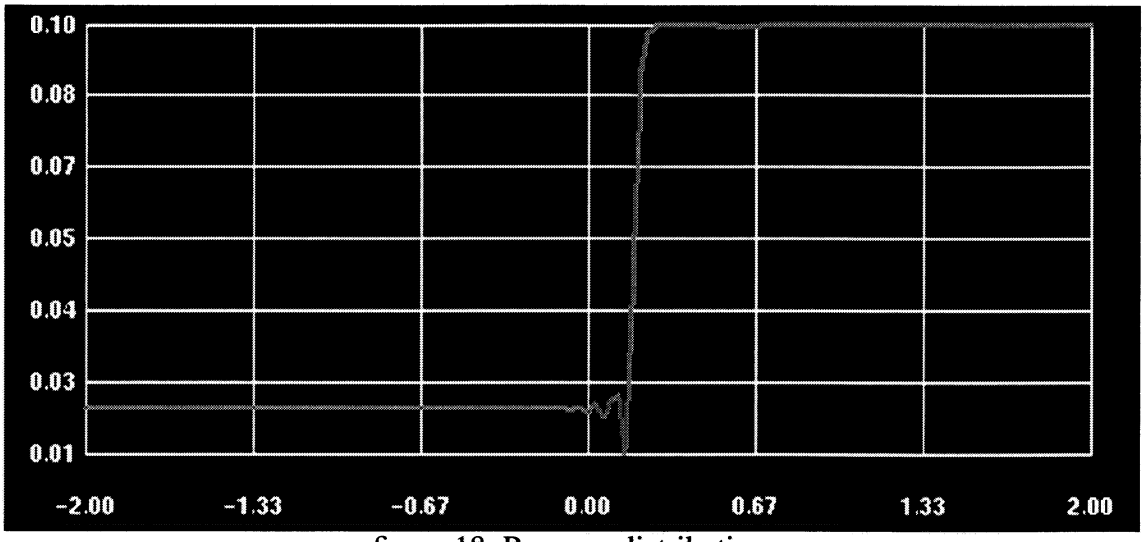


figure 18: Pressure distribution

The following figure is a plot of the shock scalar with the isentropic transient correction.

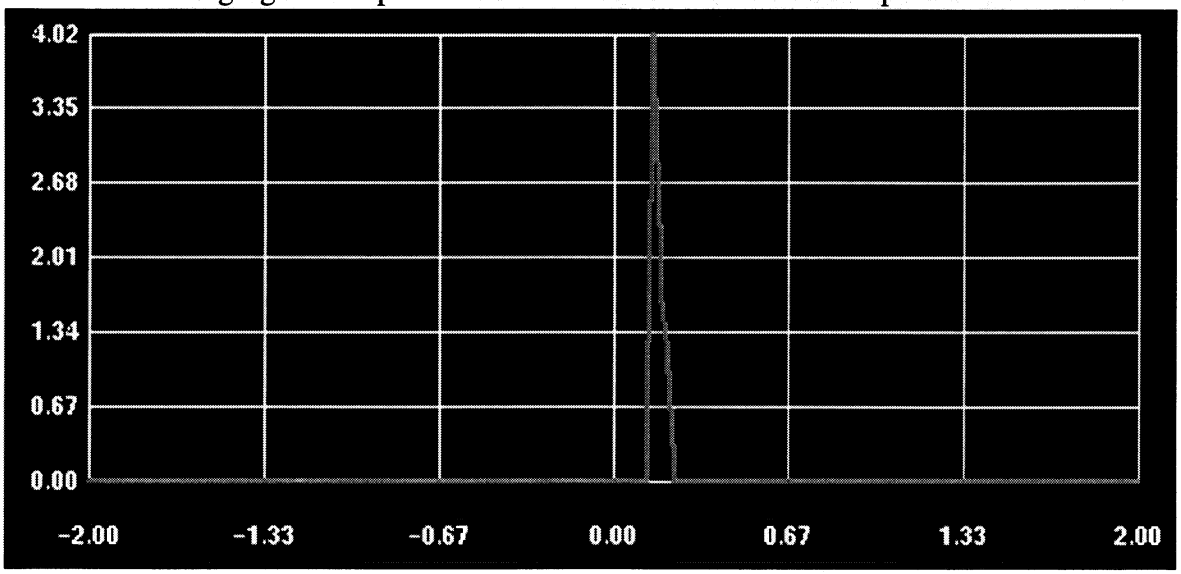


figure 19:Shock scalar

The following figure is a plot of the same quantity after a few more iterations. The shock is

clearly moving to the right, as expected.

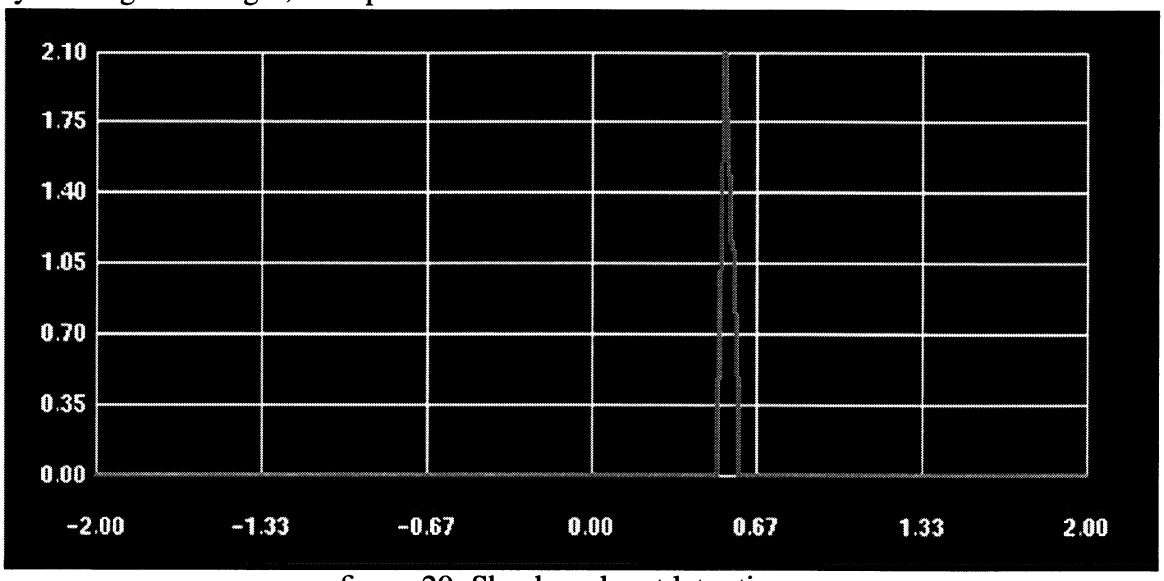


figure 20: Shock scalar at later time

The results of this experiment point to the importance of choosing a threshold value for the magnitude of the pressure gradient. Because of the slower wave speeds of the higher frequency pressure waves, oscillations in the pressure gradient take place behind the shock. These pressure gradient increases are enough to skew the results and show up in the shock detection values as a group of shocks behind the main one.

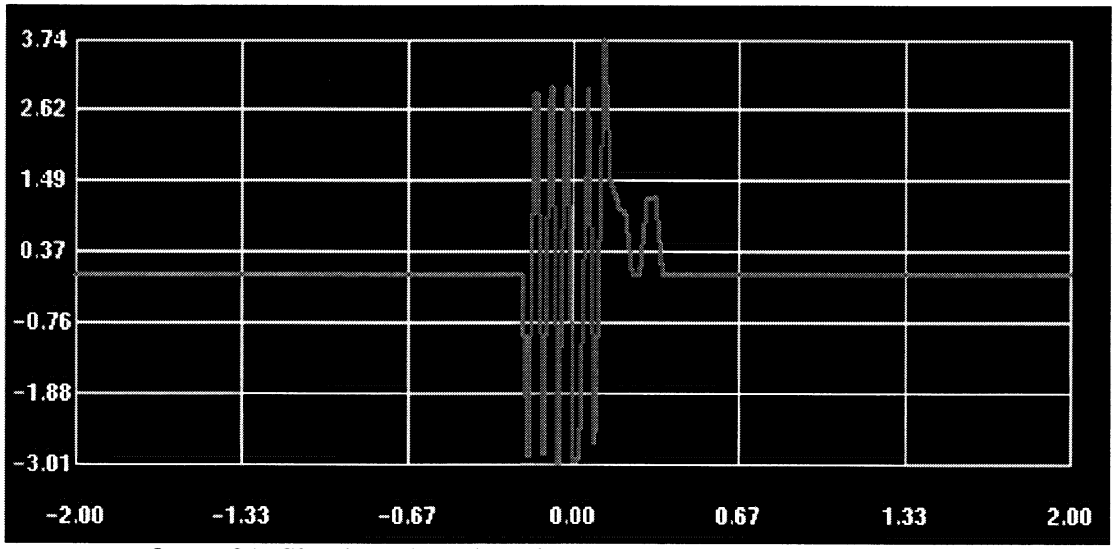


figure 21: Shock scalar when the pressure gradient is not filtered

Effect of correction term

From the above experiment, it was noted that it did not matter if the transient correction was used or not, a shock would still be indicated. This was because the upstream mach number

was greater than 1, making the normal mach number greater than one. A change in initial conditions was made to see what happens when the upstream mach number is less than 1. The upstream mach number was set to 0.9,and the pressure difference was set to 3.0, which will produce a normal shock moving to the left.

Calculation of the shock scalar

The shock test scalar was calculated with isentropic transient correction equation, that was simplified for this particular example.

$$\frac{1}{|\nabla p|}\frac{1}{a}\frac{Dp}{Dt} = -a\frac{1}{|\nabla p|}\nabla\bullet(\rho\vec{q}) + \vec{M}\cdot\hat{\nabla}p$$

The Y component of velocity was zero in this case, so the divergence term simplified to the following.

$$\frac{1}{|\nabla p|}\frac{1}{a}\frac{Dp}{Dt} = -a\frac{1}{|\nabla p|}\left(\frac{d}{dx}u\rho\right) + \vec{M}\cdot\hat{\nabla}p$$

The derivative of the pressure times the x velocity was then expanded with the product rule, yielding the final equation.

$$\frac{1}{|\nabla p|}\frac{1}{a}\frac{Dp}{Dt} = -a\frac{1}{|\nabla p|}\left(\frac{du}{dx}\rho + \frac{d\rho}{dx}u\right) + \vec{M}\cdot\hat{\nabla}p$$

The results are shown in the following two figures. In the first figure, the normal mach number is plotted against the x-axis. Notice that the normal mach number does not get above 1, which by the previous test indicates that no shock is present in the flow. However, the shock test scalar, shown in the second figure does get above 1 at the shock, indicating that the shock is indeed present. The correction term has made it possible to pick up the normal shock.

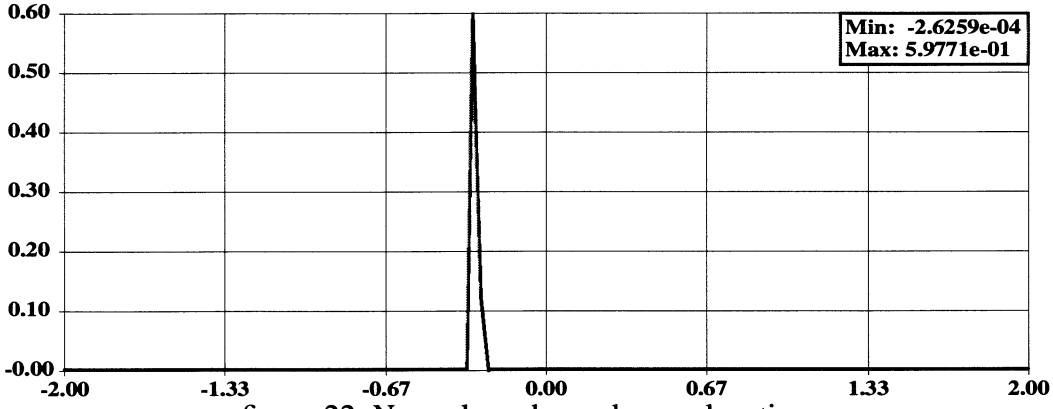


figure 22: Normal mach number vs. location

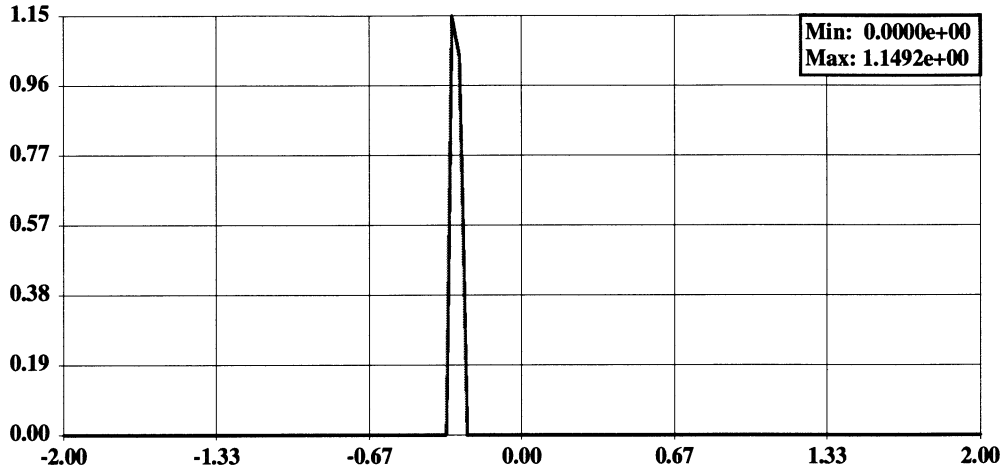


figure 23: Shock test variable vs. location

Calculation of Derived Quantities

The following quantities were needed to compute the shock scalar.

Table 2: calculated quantities

name	symbol	origin
Pressure	P	Primary output from CFD code
Velocity	U	output from CFD code
Density	rho	output from CFD code
Speed of sound	a	
Pressure gradient		Derived from the CFD output
Speed	q	vector length of velocity components
Mach vector	M	Directed along the velocity vector, with the length of the mach number
Mach number	M	Derived from CFD output and gamma

Table 2: calculated quantities

name	symbol	origin
gamma		1.4

Gradients

The pressure gradient at each node was calculated by first calculating the gradient of each cell, which was either triangular or tetrahedral, then dividing this quantity by the number of nodes in the cell, and adding it to a running sum on each node. The sum was then divided by the number of times it had been touched to come up with an average gradient for each node.

Flow solver

The flow solver used for all the test cases was a two dimensional explicit euler solver.

Conclusions

The stationary shock finding algorithm does not produce a thin shock surface that would reflect the shape of the shock in the real flow, but because of numerical smoothing, shows a shock region. Even though the shocks are thick, because the shape of the test value's contours can give the analyist information about the flow, the quality of the solution, and the quality of the mesh. If the mesh is too coarse, the shock detector will show a relatively thick shock area. If the shock is unattached, then the shock finder will show this. If the solution does not have enough dispersion, the shock finder will show this quite clearly by the nature of the contour.

The main difficulty with the algorithm is the need to choose a cutoff for the magnitude of the pressure gradient. This is especially true for transient shocks, where dispersion may cause the algorithm to display multiple non-existent shocks behind the real shock. There are a couple of ways to compensate for this effect.

Ideally, it would be nice to determine the pressure ratio across the detected shock to see if this pressure ratio were enough to produce a normal shock at Mach 1, the weakest shock possible. Any detected region not satisfying this test would then be known to not be a shock, and excluded from display.

References

1. 3D Shock Wave Visualization on Unstructured Grids Kwan-Liu Ma, John Van Rosendale, Willem Vermeer.
2. Modern Compressible Flow with Historical Perspective John D Anderson Jr. McGraw-Hill 1982.
3.